5.6 QBE

(MDU, May 2012)

Query by Example is a database query language having a two dimensional syntax. Queries in QBE are constructed using domain variables and constants in the same manner as in domain relational calculus. User needs to give an example of what

he needs to do rather than giving a procedure to obtain a desired result. User while sitting on a terminal can call one or moreable skeletons and then name the relation and attributes represented by these skeleton tables. To specify any query user needs to fill in one or more rows of the table. These tables are automatically converted into SQL by DBMS. Users do not need to remember names of the relation as they are displayed on screen. A domain variable is always preceded by an underscore_. The constant specified should exactly match with the value in the column of the table. A P. is used for print command.

Example: Consider the relation given in Chapter 4 and answer the following queries:

1. Display all the customer information

customer	c_name	c_city	c_state
P.			

By writing P. in customer (table name) will print every row and column of customer table. We can also write P. in every column.

2. Find account number of all balances greater than Rs. 50,000.

account	acc_no	b_Id	balance
	P.		>50,000

3. Find all customers who live in same city as Rajesh

customer	c_name	c_city	c_state
	P._x	_y	
	Rajesh	_y	

In above query we have to customer name based on some condition so P. is used in c_Name attribute. Suppose city of Rajesh is y. Now we have to find all customer x whose city is y. Therefore –y is used as city corresponding to customer names –x and Rajesh.

4. Find names of the customers who have loan at branch with branchId 206.

loan	l_no	b_Id	amount
	_x	206	

borrow	c_name	l_no
	p._y	_x

In above query, l_no of loan should match l_no of borrow. So, in l_no field of both table skelton same variable x is used.

5. Find names of customers who have an account but do not have any loan.

deposit	c_name	acc_no
	p._x	

borrow	c_name	l_no
¬	_x	

Consider Box, it is used to specify constraints on domain variable.

6. Find all loan number with loan amount greater than Rs.50,000 but less than Rs.90,000

account	acc_no	b_Id	balance
	P.		_x

Conditions
_x > 50,000
_x < 90,000

In the above query *Condition Box* can be used to specify the constraints on domain variable.

Result Relation

When result of a query include attributes from many relations then the result relation can be used.

Example: Find customer name and account number for all account in branchId 206.

account	acc_no	b_Id	balance
	_x	206	

deposit	c_name	acc_no
	_y	_x

result	c_name	acc_no
P.	_x	_y

Ordering

QBE provides mechanism for organising the data in ascending or descending order. AO and DO command are used to arrange data in ascending and

descending order respectively. Eg Display all customer name arranged in ascending order:

customer	c_name	c_city	c_state
	P.AO		

Aggregate Operations:

QBE provides various built in function to perform function like average, sum, maximum, minimum, etc. Some functions are:

CNT, AVG, MAX, MIN, SUM

Example: Find average balance of all accounts

account	acc_no	b_Id	balance
			P.AVG,
			ALL._x

Example: Find total number of customers having an account in bank.

customer	c_name	c_city	c_state
	P.CNT.UNQ.ALL		

Creation of Relation:

A new relation in QBE can be created by using I symbol.

For example, account relation can be created y typing following entries in skeleton.

I.account	I.	acc_no	b_Id	balance
KEY	I.	Y	N	N
Type	I.	CHAR (10)	CHAR (5)	FLOAT
DOMAIN	I.	NUMBER	BRANCH	AMOUNT
INVERSION	I.	N	N	N

Symbol I. indicate an insertion operation. The first I. in first line of skeleton table is for insertion of relation and second I. denotes the insertion of attributes of the relation.

Rest rows indicate the insertion of characteristic of attributes by specifying various keywords. The meaning of these keywords are stated as follows:

(i) **KEY** – It indicates whether an attribute is a key or a part of a key indicates that column is a key or part of a key.

(ii) **TYPE** – It indicates data type. Following data types are supported by QBE.
- CHAR – single character
- CHAR (n) – n character
- FLOAT – floating point number
- FIXED – integer

(iii) **DOMAIN** – It indicates domain name of an attribute. If a domain variable appear in more than one column in a skeleton table than corresponding columns must have same domain name.

(iv) **INVERSION** – It indicates whether a secondary index or attribute is to be created or not.

Insertion

Insertion of a tuple can be done by I. symbol.

Example: Add a new customer to customer relation

customer	c_name	c_city	c_state
I.	John	Ajmer	Rajasthan

Example: Add a new customer to customer relation with c_name Manas, c_city Jaipur and c_state same as c_state of John

customer	c_Name	c_City	c_State
I.	Manas	Jaipur	_x
	John		_x

Deletion

Deletion of a tuple in QBC can be done by D. Symbol.

Delete tuple from customer where c_name is John

customer	c_name	c_city	c_state
D.	John		

Updation

U. symbol can be used to update any tuple. Eg. Increase the balance in every account by 2%.

account	acc_no	b_Id	balance
U.			_x
			_x*1.02

Condition Box can also be used to specify condition during update operations.